

# **The Future of Semantic Forms**

Yaron Koren

SMWCon Spring 2010

May 23, 2010

Semantic Forms has grown, over three years, from a utility for editing template calls into a complex "framework".

This has led to a variety of issues and challenges...

1) Too much code for one extension?

2) SF contains elements that don't naturally belong there.

Namely:

- Special>CreateProperty
- Special>CreateTemplate
- Special:Templates
- #arraymap
- #arraymaptemplate

3) Tightly integrated with SMW - but maybe it should work without SMW?

4) For admins, maintaining templates, forms and categories together is a challenge.

Almost any change in data structure requires modifying two to four pages.

5) No separation of "model" and "view" - data and display are defined in the same place

6) Non-web applications, like iPhone apps, may want to create their own interfaces to the wiki, using form definitions



7) Form-based editing is (hopefully) coming to Wikipedia!

Javascript-based, and uses its own XML format for defining template structures.

Should SF be able to read this format? Or even switch to it?

8) Plus all the usual bugs and feature requests

The solution for all these issues is...

???

Split up of extension might make sense.

For instance: Page Object Model (POM)  
extension can handle SF's reading and writing  
of template calls.

One potential piece of the puzzle: a new  
“Semantic Classes” extension

# Classes

We can think of a certain data type as a “class”, and the set of wiki pages for that type – category, form, template(s), properties, filters – as the “class pages”.

## Semantic Classes extension

Would allow for defining all the information about a "class" - for display, form, and drilldown - within XML contained within a category page.

This XML could then be used to generate template, property, form and filter pages.

The XML could then also be edited via some kind of helper tool.

# Special pages

Probably two:

- `Special:EditClassXML`
- `Special:GenerateClassPages`



# Sample XML

In a page called “Category:Cities”:

```
<SemanticClass>
  <Name>City</Name>
  <Form>City</Form>
  <Field>Population</Field>
    <SemanticProperty>
      <Name>Has population</Name>
      <Type>Number</Type>
    </SemanticProperty>
    <FormInput>
      <InputType>text</InputType>
      <Size>20</Size>
    </FormInput>
    <Filter>
      <Label>Population</Label>
    </Filter>
  </Field>
  ...
</SemanticClass>
```

# Customization

Once templates, etc. are generated, they can be modified by users - they would just need to be maintained by hand, since running "generate" again would overwrite the changes.

On the other hand, for basic implementations, it may be possible to do away with form-definition and filter pages altogether - Semantic Forms and Semantic Drilldown could read the XML directly.

# Advantages of Semantic Classes approach

- Single point where all data for a "class" is stored
- Separation of data structure from display (mostly - the one exception is the ordering of fields)
- Class can be edited, not just created, via a helper form - for basic data structures, admins may never have to edit wiki text

# More advantages of Semantic Classes approach

- Could enable “inheritance” - classes could inherit from other classes, and only have to define their differences from the parent class
- Could allow import/export from other data-definition formats, like OWL, UML and Wikipedia “TemplateInfo” XML

Anybody up for funding this thing?

If so, contact me. :)